

# A Framework to Analyze, Compare, and Optimize High-Performance, On-Board Processing Systems

Nicholas Wulf, Alan D. George, and Ann Gordon-Ross

NSF Center for High-Performance Reconfigurable Computing (CHREC)

Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611

{wulf, george, ann}@chrec.org

**Abstract**—On-board processing systems are often deployed in hostile environments and must therefore adhere to stringent constraints such as low power, small size, and high dependability in the presence of faults. Since it is challenging for designers to simultaneously consider the many design tradeoffs and meet the numerous and unique demands and constraints of various on-board systems, designers typically rely on a limited set of familiar devices and design strategies that may not be optimal for a particular system’s operating situation. In this paper, we present a framework to ease these system design challenges and aid designers in considering a broad range of devices and strategies for on-board processing, highlighting the most promising options early in the design process. Our framework considers the interactions between four key system properties—device, mission, fault-tolerant strategy, and application—which allows the framework to evaluate how well a design will meet mission constraints based on design evaluation metrics to identify tradeoffs between varying devices and fault-tolerant strategies. This paper focuses on the power and dependability evaluation metrics, which our framework calculates and leverages to evaluate the effectiveness of varying system designs. Finally, we use our framework to evaluate system designs for two case studies on hyperspectral-imaging (HSI) missions.

exploring farther-reaching areas, transmission latencies can be on the order of tens of minutes or more, which hinders remote systems that rely on real-time operating decisions from a home-base station.

In order to address increasing bandwidth pressure and transmission latencies, remote systems include on-board processing capabilities to process the raw data in-situ and transmit only the smaller, processed data. Additionally, on-board processing empowers remote systems to perform the necessary calculations for making intelligent autonomous operating decisions in real-time, thereby reducing the need for high-latency operation instructions from a distant home-base station.

However, incorporating on-board processing into an aerospace mission is challenging when considering the often stringent size, weight, and power (SWaP) constraints. Power is generally the most limiting of these constraints since power is difficult to harvest and store, and increasing the processing performance increases the power consumption. Challenges in aerospace also include radiation effects, which cause unexpected and erroneous behaviors in processing systems and are exacerbated by decreasing feature sizes and an increasing number of processing cores. Therefore, once a designer has defined an aerospace mission’s system platform, environment, and applications (e.g., hyperspectral imaging (HSI), real-time landing, obstacle avoidance, etc.), the primary design challenge is device and fault-tolerant (FT) strategy selection. The device must perform well with the mission’s applications and be capable of operating well in the mission’s environment. An appropriate FT strategy will also be necessary for most missions in order to guarantee correct operations without requiring excessive resource overhead.

A successful on-board processing system design meets or exceeds all mission constraints (e.g., maximum power usage, maximum fault rate, minimal processing throughput, etc.). Since there are tradeoffs between these mission constraints, the set of successful designs contains many Pareto-optimal designs [1]. Therefore, the designer must choose the best design based on the mission constraints and acceptable tradeoffs. For example, since mission failure may be catastrophic (e.g., loss of life), a designer may trade lower power consumption for lower fault rates. Alternatively, missions that may be updated and altered after deployment may trade lower fault rates for reduced

## TABLE OF CONTENTS

<b>1. INTRODUCTION .....</b>	<b>1</b>
<b>2. BACKGROUND AND RELATED WORK.....</b>	<b>2</b>
<b>3. FRAMEWORK .....</b>	<b>3</b>
<b>4. EVALUATION METRICS .....</b>	<b>5</b>
<b>5. CASE STUDY ANALYSIS .....</b>	<b>7</b>
<b>6. CONCLUSIONS .....</b>	<b>12</b>
<b>ACKNOWLEDGMENTS.....</b>	<b>12</b>
<b>REFERENCES.....</b>	<b>12</b>
<b>BIOGRAPHIES.....</b>	<b>14</b>

## 1. INTRODUCTION

Unmanned, remote-sensing systems are commonly used in aerospace and outer-space to sense and collect raw data from the surrounding environment. The collected data is typically transmitted to a central home-base station where high-performance processing systems process and analyze the data. However, rapidly improving sensor technology has significantly increased the amount of collected data, which may exceed the remote system’s transmission bandwidth. Additionally, since remote systems are continually

power or device utilization to enable the processing payload to be increased after deployment. Not only is determining the best design a complex task, the design exploration space is often limited by the designer’s reliance on familiar devices and FT strategies and development time constraints (i.e., designers may not have time to explore new devices and FT strategies). These limitations narrow the design space’s scope, possibly resulting in successful yet non-Pareto-optimal designs.

Designers evaluate system designs using evaluation metrics, such as power, dependability, device utilization, mission lifetime, and design cost. Since power and dependability are often the most critical evaluation metrics for on-board processing systems in constrained and hostile environments, such as aerospace, our work focuses on these two metrics. The power metric measures how much power the processing device will consume during the mission. The dependability metric quantifies a system’s ability to correctly operate within the mission environment and is often represented by the mean time to failure (MTTF), mean time between failures (MTBF), or data loss rate.

To aid designers in addressing the challenges of on-board processing system design, we present a novel framework that determines a set of Pareto-optimal device and FT strategy combinations based on a mission’s constraints. Our framework considers four key system properties: mission, application, device, and FT strategy. The designer specifies the mission and application properties. The mission property defines information about the mission environment and dictates the resources and constraints of the on-board processing system based on design constraints and available platform resources (e.g., sensors, power generation, memory capacity). The application property defines the on-board processing tasks, which are typically sensor data processing and autonomous operation decisions (i.e., autonomy processing). Once these system properties have been defined within the framework, the framework analyzes these properties with respect to varying devices and FT strategy combinations to produce power and dependability metrics data to determine the Pareto-optimal system designs.

Our framework provides several designer benefits: 1) alleviates challenges associated with designing on-board processing systems for aerospace; and 2) produces system evaluation metrics data, allowing designers to quickly select the best design by comparing tradeoffs between the Pareto-optimal designs, even if the designer is not yet familiar with the devices or FT strategies.

The remainder of this paper is organized as follows. Section 2 discusses the background and related work that provides the foundation for our framework. Section 3 presents an overview of our framework and Section 4 discusses the framework’s evaluation metrics. In Section 5, we present two case studies of HSI missions to demonstrate the framework’s use and effectiveness.

## 2. BACKGROUND AND RELATED WORK

Our framework leverages previous work related to each of the four key system properties and introduces a novel evaluation methodology that combines these properties together, producing evaluation metrics data to assess and compare various on-board processing system designs.

Pease et al. [2] discuss acceptable device selection based on an environment’s varying radiation levels. A device database stores radiation data for a set of known devices and allows designers to quickly eliminate unacceptable devices. Our framework leverages a similar device database to store radiation data, with additional data on the device’s processing capabilities and power consumption. Williams et al. [3] define a general methodology for determining the maximum processing capabilities of a given device, referred to as the computational density (CD). Their methodology considers a wide range of device architectures (e.g., CPU, DSP, FPGA, GPU) and considers operation types as well as precision when calculating the CD. Our framework leverages this CD methodology to create device processing capability data for the framework’s device database.

For Earth-orbiting missions, CREME96 [4] predicts the average radiation-flux values experienced by a processing system due to the surrounding environment. Using user-provided radiation data for specific devices, CREME96 also predicts device upset rates based on the radiation-flux effects. Due to the effectiveness and accessibility of CREME96, our framework leverages CREME96’s comprehensive analysis to determine device upset rates for Earth-orbiting missions.

FT strategies increase software and hardware fault tolerance using redundant calculations and/or information, which allows processing systems to operate correctly despite effects caused by upset-inducing radiation. However, this redundancy incurs processing and/or area overheads, which increase as the FT strategy’s fault-mitigating capabilities increase. For example, replication-based FT strategies, such as triple-modular redundancy (TMR) [5] and single-error correction with double-error detection (SECDED) [6] codes, are capable of detection and correction, and incur  $\sim 200\%$  and  $(100 \cdot \log_2(n)/n)\%$  area overheads, respectively. Application-dependent FT strategies can offer lower overhead fault-mitigating capabilities, such as algorithm-based FT (ABFT) [7], which leverages the linear properties of common matrix operations to produce checksums that detect errors in the final calculated matrices. Device-dependent FT strategies, such as reconfigurable FT (RFT) [8] and adaptive FT [9], leverage an FPGA’s reconfiguration capabilities and the time-varying nature of orbital radiation to dynamically increase/decrease the fault-mitigating capabilities. Our framework considers a wide range of FT strategies, which allows designers to evaluate FT strategies with respect to the specific application and device and consider tradeoffs between the fault-mitigating capability and performance/area overhead.

In order to select the Pareto-optimal designs, it is important to understand how the application affects a device's performance. For example, FPGAs are effective for bit-level and fixed-point operations, but less effective for double-precision, floating-point operations due these operations' much higher utilization of reconfigurable resources. Asanovic et al. [10] address this issue for high-performance computing (HPC) systems by identifying 13 common kernels that represent the essential operations of nearly all HPC applications. This subsetting enables HPC system designers to quickly and effectively study a broad range of applications and application behaviors with little loss of accuracy. Our framework leverages this subsetting methodology to identify the most common kernels that represent the majority of all on-board processing applications, which allows our framework to analyze a broad range of on-board processing applications without requiring separate, specific research into each application.

### 3. FRAMEWORK

Our framework determines the Pareto-optimal set of device and FT strategy combinations based on the four key system

properties, allowing designers to select the best design based on desired tradeoffs, regardless of a designer's familiarity with the devices and FT strategies. Although this paper focuses on FPGA devices for aerospace environments, our framework includes a wide range of devices (e.g., CPUs, DSPs, FPGAs, GPUs) as well as a diverse set of environments (e.g., outer-space, aerospace, underwater) and is easily extendable to additional devices and environments.

The remainder of this section is organized as follows. Section 3-1 presents an overview of our framework, focusing on overall scope, general concepts, and the framework's components, and Section 3-2 provides details on each of these components.

#### 3-1. Overview

Figure 1 depicts an overview of our framework, which is composed of five components. The first four components are system property components, which include the device set, the mission characteristics, the FT strategy set, and the application kernel set components and correspond to the four key system properties (device, mission, FT strategy, and application, Section 1), respectively. The fifth

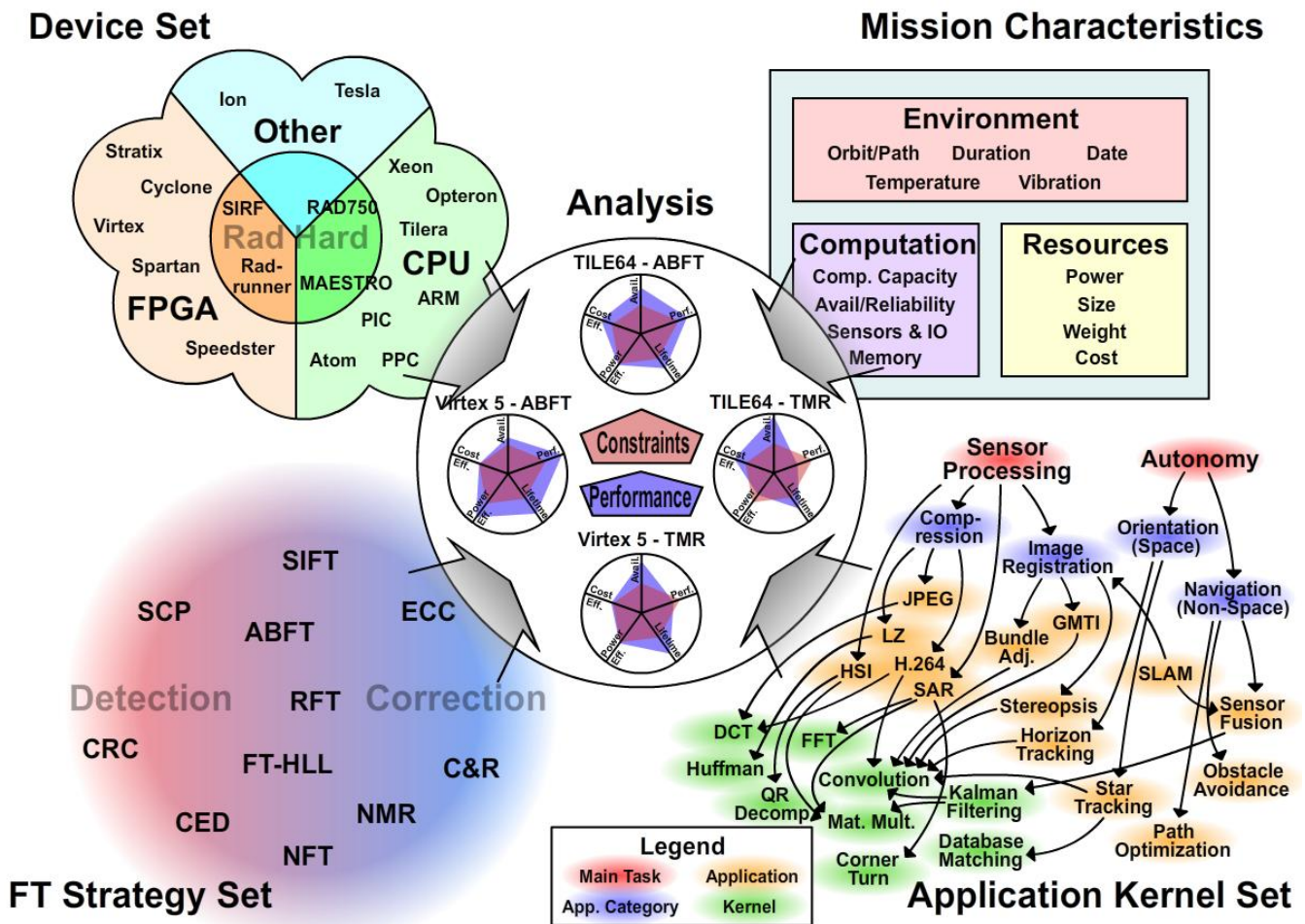


Figure 1 – Framework overview consisting of the four system property components (corners) and analysis component (center)

component, the analysis component, corresponds to the evaluation metrics (power and dependability, Section 1).

The system property components consist of both designer-specified data and research data obtained from literature. The designer provides the mission characteristics, since the framework cannot have a priori knowledge of the system platform, environment, and constraints. Our framework pre-defines the device set, FT strategy set, and application kernel set based on literature research data (Section 3-2).

The analysis component combines the system property components' data and produces evaluation metric data, which the designer evaluates to select the best design. Each evaluation metric combines the system property components' data in a unique method based on the specific evaluation metric's dependency on the system property components' interactions. For example, the power metric evaluates device performance with respect to an application, whereas the dependability metric evaluates device radiation response data with respect to the mission environment. Alternatively, the dependability metric evaluates the FT strategies' fault-mitigation capabilities, whereas the power metric evaluates the FT strategies' performance and area overheads. Finally, evaluation metrics only evaluate valid designs that use device- or application-dependent FT strategies with the corresponding devices and applications.

### 3-2. Components

The device set contains data on a broad range of device architectures (e.g., CPU, DSP, FPGA, GPU) as well as any available radiation-hardened versions of these devices. The device set's data records three characteristics for each device: power range; processing capability; and radiation response. The power range defines the device's minimum and maximum power usage depending on resource utilization (e.g., an FPGA design that uses almost no device logic would consume the minimum power, while an FPGA design that uses the maximum amount of a device's logic would consume the maximum power). The processing capability is represented using the CD methodology [3] and depends on the type and precision of the application's operations. The radiation response typically involves determining a device's linear energy transfer curve, which represents the likelihood of a single particle disrupting the device for varying levels of particle energy. Literature research provides the radiation response data, since this data is sufficient for the framework's analysis, and obtaining this data via experimental analysis is difficult and time-consuming.

The mission characteristics define the mission environment, available resources, and computational constraints. The mission environment includes data on the mission's specific path (e.g., an orbit in space or a route along the ocean floor), the mission's duration (e.g., months or years), the mission start date for considering time-dependent environments, and any other hostile conditions that must be considered (e.g., extreme temperatures or excessive vibration). The available

resources include the SWaP restrictions and may also include a defined monetary budget for designing and building the system. The framework uses the constraints defined in the resource data to test the successfulness of various designs. The computational constraints dictate the acceptable fault rates, required processing throughput based on the incoming sensor data's throughput, and the maximum allowable memory usage based on on-board memory constraints.

The FT strategy set contains literature research data on the most effective and/or common FT strategies, which includes a wide variety of FT detection and/or correction strategies, some of which are device- or application-dependent. The FT strategy set records three characteristics for each FT strategy: effectiveness; overhead; and dependencies. The effectiveness is the FT strategy's fault-mitigation capability (e.g., detection only, or detection and correction). For example, if a non-fault-tolerant (NFT) system has a 1% chance of experiencing a fault, adding a TMR FT strategy to the system will correct 98% of these faults. The overhead refers to the extra processing that all FT strategies require due to redundant calculations (e.g., ~200% overhead for TMR). Finally, the dependencies define which devices or applications correspond to a given FT strategy, ensuring that the framework only evaluates valid designs.

The application kernel set contains the subset of common kernels (e.g., matrix multiplication and fast Fourier transform) representing the essential operations of the vast majority of on-board processing applications. Identifying the common kernels (or applets) is a key challenge and important area of research for our framework. Research involves analyzing a comprehensive survey of aerospace applications with the goal of identifying the smallest subset of common kernels that encompasses the largest amount of the applications' constituent kernels. If future analysis determines that emerging aerospace applications are not necessarily covered under the current subset of kernels, the subset can easily be expanded to include these new kernels.

In addition to mapping applications to one or more of these kernels, our framework categorizes applications as either sensor processing or autonomy processing. Sensor processing is the processing of the raw data collected from on-board sensors with the purpose of compressing and/or extracting important information before transmission. Autonomy processing is the ability of the on-board processing system to make intelligent decisions and take effective action based solely upon in-situ analysis of the environment, such as circumnavigating obstacles and locating landing zones. Sensor processing typically focuses on meeting transmission throughput constraints, while autonomy processing focuses on reliably meeting real-time deadlines.

The analysis component's evaluation metrics set contains the functions that calculate the evaluation metric data from the system property components data. From the evaluation metric data, our framework determines the valid, successful

designs. Figure 1 depicts four potential designs—TILE64-ABFT, TILE64-TMR, Virtex 5-ABFT, and Virtex 5-TMR—and the designs’ attained performance (larger blue pentagon) and constraints (defined in the mission characteristics, smaller red pentagon) for each evaluation metric. A design is successful if the attained performance meets or exceeds the constraints (i.e., the blue pentagon encompasses the red pentagon). Finally, our framework outputs evaluation metrics data for the Pareto-optimal designs, from which the designer can easily determine the best design based on mission constraints and desirable metric tradeoffs.

#### 4. EVALUATION METRICS

The analysis component of our framework combines data from the four key system properties into a concise set of evaluation metrics, providing designers with a quick and valuable insight into a variety of designs. This paper focuses on power and dependability, our framework’s first and foremost evaluation metrics for aerospace missions.

The remainder of this section is organized as follows. Section 4-1 presents the process for calculating the amount of power consumed by a design’s device. Section 4-2 presents the process for calculating a design’s dependability.

##### 4-1. Power

Figure 2 depicts the power metric calculation process. First, our framework calculates the system’s required processing in terms of type and rate of operations performed based on the designer-specified application processing and sensor input-data rate. For example, consider a simple on-board image-processing system that uses a camera to capture images of Earth from space with a sensor data rate of three images per second, four megapixels per image, and three 8-bit color channels (i.e., red, green, blue) per pixel. The system sums each pixel’s three color values to an aggregate sum to determine if the average brightness of the image exceeds a certain threshold. Since adding multiple 8-bit values produces a result larger than 8 bits, the system processing can be summarized as three 16-bit addition operations per pixel, which is a required processing of 36 million 16-bit addition operations per second.

Our framework uses the required processing result and device CD (determined by the methodology of [3]) to calculate device utilization, which is the amount of device resources a system uses relative to the total amount of device resources available. 100% device utilization means that the system is using the device at the device’s maximum potential. Our framework calculates device utilization as the ratio of the required processing to the device’s CD. This CD value must correspond to the type and precision of operations used in the required processing. For the image-processing example and a sample device with a 16-bit integer addition CD of  $10^8$  operations per second, the device utilization is 36%.

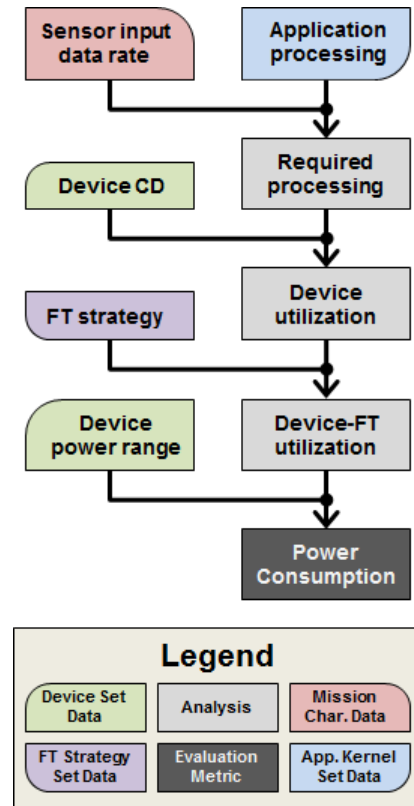


Figure 2 – Power metric calculation

Device-FT utilization updates the device utilization to include the FT strategy’s area overhead based on Equation 1. For the image-processing example and a TMR FT strategy, for instance, TMR introduces a ~200% overhead, which results in a device-FT utilization of 108%. Since the device-FT utilization is greater than 100%, the system either requires more than one device or a different device with greater resources.

$$Util_{dFT} = Util_{device} \cdot (100\% + Overhead_{FT}) \quad (1)$$

The final output of the power metric calculation is the system’s total power consumption, which is calculated based on the device’s power range (minimum and maximum) and the device-FT utilization value. A device’s static power primarily influences the minimum power, which is measured as the minimum power required for the device to be powered on and corresponds to 0% device utilization. The device’s maximum power corresponds to 100% device utilization. Equation 2 calculates the device’s power consumption for any device-FT utilization between 0% and 100% using linear interpolation between the minimum and maximum power. If device-FT utilization is greater than 100%, Equation 3 determines the number of required devices  $n$ , and Equation 4 calculates the total consumed power for all  $n$  devices. For this image-processing example and for a sample device with minimum and maximum power of 5 Watts and 15 Watts, respectively,

since the device-FT utilization is 108%,  $n$  is 2 and the total power consumed is 20.8 Watts.

$$P_{\text{single}} = P_{\text{min}} + Util_{\text{dFT}} \cdot (P_{\text{max}} - P_{\text{min}}) \quad (2)$$

$$n = \left\lceil \frac{Util_{\text{dFT}}}{100\%} \right\rceil \quad (3)$$

$$P_{\text{total}} = \sum_{i=1}^n \left( P_{\text{min}} + \frac{Util_{\text{dFT}}}{n} (P_{\text{max}} - P_{\text{min}}) \right) \\ = nP_{\text{min}} + Util_{\text{dFT}} \cdot (P_{\text{max}} - P_{\text{min}}) \quad (4)$$

#### 4-2. Dependability

Figure 3 depicts the dependability metric calculation process. First, our framework requires literature research data for both the environmental radiation and the device’s radiation response. Environmental radiation data describes the particle flux (i.e., particles per square meter per second) for varying linear energy transfer (LET) values. LET measures the amount of energy deposited by a particle as it passes through an object (silicon in this case). Figure 4 depicts example environmental radiation data obtained from [11]. In this example, a square meter of silicon will experience a  $10 \frac{\text{MeV}}{\text{mg}\cdot\text{cm}^2}$  LET particle every second, a  $40 \frac{\text{MeV}}{\text{mg}\cdot\text{cm}^2}$  LET particle every 30 years, and less than one  $100 \frac{\text{MeV}}{\text{mg}\cdot\text{cm}^2}$  LET particle every 3 millennia.

The device radiation response data describes the cross-section of the vulnerable device area that experiences an upset (i.e., bit flip) when hit with a particle of a certain LET level. For example, a device with an area of  $400\text{mm}^2$  has an effective vulnerable area of  $40\text{mm}^2$  if only 10% of particles

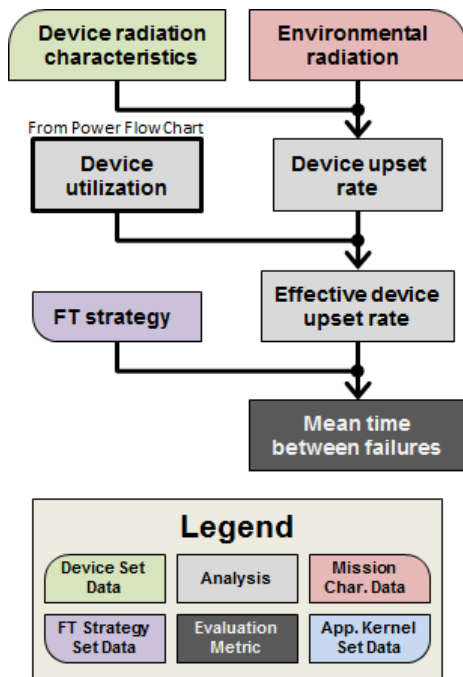


Figure 3 – Dependability metric calculation

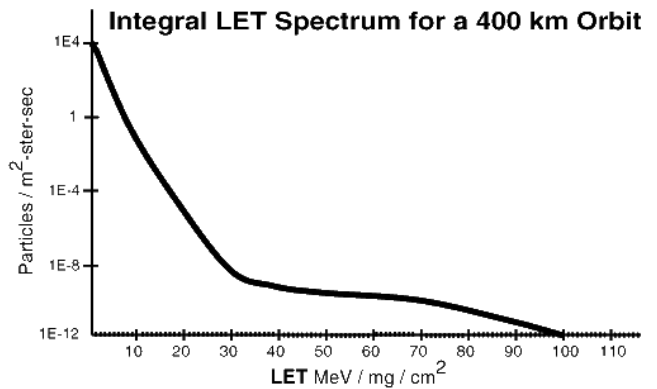


Figure 4 – Example environmental radiation data [11]

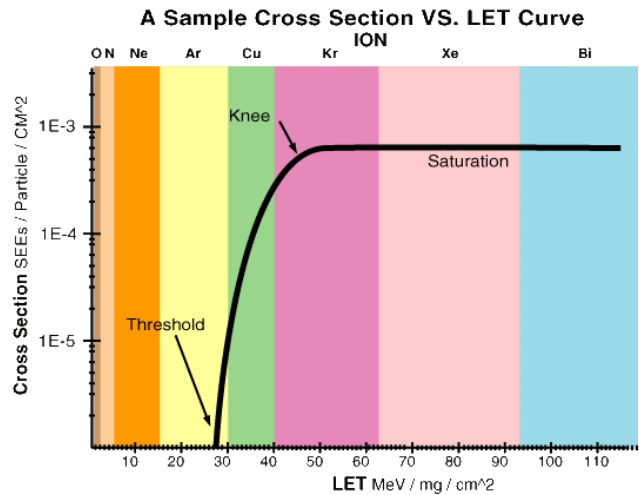


Figure 5 – Example device radiation characteristic data, with colored regions corresponding to the specific element used to test an LET range [11]

hitting the device cause an upset. Literature research typically presents cross-section values in units of  $\text{cm}^2/\text{device}$  or  $\text{cm}^2/\text{bit}$ . Figure 5 shows a Weibull curve for example device radiation response data, which is characteristic of LET data for all devices [11]. The defining regions of the Weibull curve are the threshold, knee, and saturation region. The threshold defines an LET value, below which particles do not deposit enough energy to cause an upset. In the region between the threshold and the knee, particles start depositing enough energy to potentially upset the device. The saturation region begins at the knee and remains at a constant saturation cross-section value for increasing LET values. The saturation cross-section value corresponds directly to the vulnerable area of the bits. Increasing LET beyond the knee has no effect, since the knee LET already deposits enough energy to cause an upset 100% of the time when the vulnerable area is hit.

Our framework calculates the device upset rate based on the rate and effect of the various particles on the device, the values for which are found in the environmental radiation data and device radiation response data. The device upset rate measures the rate at which upsets occur in the whole

device, including regions of the device that may not be used. If upsets occur in the unused resources of the device, the upsets have no effect on the overall system since any output from the unused resources is ignored by the design. Therefore, the effective device upset rate is calculated as the product of the device upset rate and the device utilization (Section 4-1), which measures the relative amount of device resources used.

With the effective device upset rate and the applied FT-strategy, our framework calculates the MTBF, which quantifies the average time a device can operate without experiencing a failure. MTBF is calculated differently for different FT strategies, which may include variables such as non-FT effective device upset rate and input data size. For example, Equation 5 calculates a TMR [5] system’s reliability, which is the probability that there is no system upset for some unit of time. If a non-TMR system has a reliability of 99.0% after one day, then TMR raises the reliability to 99.97%, protecting against 97% of the upsets as compared to the non-TMR system. Conversely, if the non-TMR system has a reliability of 80.0%, TMR raises the reliability to 89.6%, protecting against less than half of the upsets. For other FT strategies, it may not be possible to realistically calculate the FT strategy’s fault-mitigating capabilities, requiring either fault-injection testing or literature research. After calculating the final upset rate for the system, our framework calculates the MTBF by inverting the upset rate.

$$R_{TMR} = 3R_{Orig}^2 - 2R_{Orig}^3 \quad (5)$$

## 5. CASE STUDY ANALYSIS

This section introduces two currently deployed HSI missions, which serve as case studies for testing and analyzing our framework’s methodology. Section 5-1 introduces HSI data collection and materials analysis as well as our two case-study missions. Section 5-2 details our framework’s calculation process for the power and dependability metrics for a Virtex-4 with ABFT for both case-study missions. Section 5-3 discusses the power and dependability metrics of 18 different designs for both case-study missions.

### 5-1. Experimental Setup

HSI sensors and conventional color cameras gather information about a scene by measuring the energy intensity of various electromagnetic spectral bands. Conventional color cameras have a spectral range that covers the visible spectrum and divides this range into three spectral bands: red, green, and blue. However, HSI sensors typically use over 100 spectral bands to cover spectral ranges from the visible to the infrared spectra. When imaging a scene, a conventional camera produces three simultaneous images (one for each spectral band), whereas an HSI sensor produces hundreds of simultaneous images. Stacking these

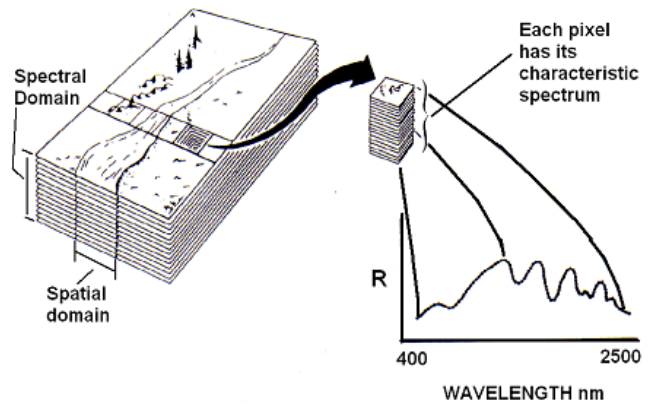


Figure 6 – HSI image cube and the characteristic spectrum of a single pixel [12]

images together forms a three-dimensional image cube for the HSI sensor, where the two spatial dimensions designate an image pixel, and the spectral dimension designates a specific spectral band.

HSI analysis attempts to identify certain materials within a scene by comparing known material spectral signatures with observed characteristic spectrums. As shown in Figure 6, a pixel’s characteristic spectrum is the group of data from each spectral band that corresponds to the given pixel. A priori measurements produce spectral signatures for any materials of interest, which define the material’s reflectance values for the spectral bands used by the HSI sensor. Figure 7 shows the spectral signatures for a few example materials. By comparing each pixel’s characteristic spectrum to the set of material spectral signatures, HSI analysis can identify any material of interest and the material’s locations in the observed scene, producing an output image similar to Figure 8. Humans employ an analogous real-world analysis process when people use the color of an object to determine the object’s material composition (e.g., brown on an apple indicates rotting). The greater spectral detail provided by

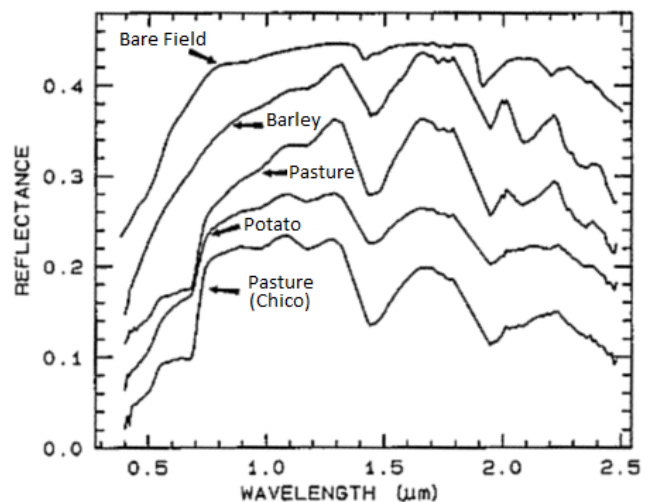
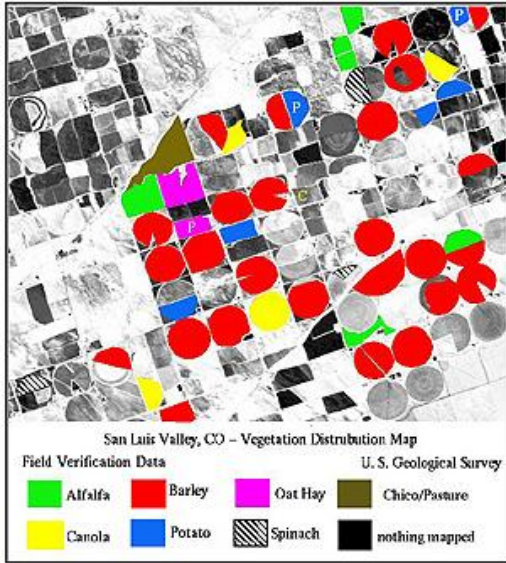


Figure 7 – Spectral signatures for various materials [13]



**Figure 8 – HSI analysis identifying various kinds of vegetation [13]**

HSI sensors enables HSI analysis to more precisely identify materials (e.g., distinguishing between different types of green vegetation).

Remote HSI imaging systems typically transmit collected image cubes to a ground station where high-performance processing systems perform HSI analysis. However, advances in space-borne electronics and improvements in fault-mitigating technology enable on-board HSI analysis, which provides several advantages, such as enabling the HypIRI [14] HSI system to provide real-time critical information on natural disasters (e.g., volcanoes, wildfires, and drought). HSI analysis also reduces image cubes to around 1% of the cube’s original size, affording more efficient data storage and transmission.

Assessing the feasibility of an HSI on-board processing system requires estimation of the processing required for the HSI analysis on the incoming sensor data. However, since around 97% of the required processing involves a single matrix-multiply operation, these estimations are simplified. The matrix-multiply operation involves calculating the autocorrelation sample matrix  $\mathbf{R}_{L \times L} = (\mathbf{A}_{N \times L})^T (\mathbf{A}_{N \times L})$ , where  $N$  is the number of pixels and  $L$  is the number of spectral bands. Matrix  $\mathbf{A}_{N \times L}$  represents the sensor’s image cube since spectral data for each pixel corresponds to a certain row of  $\mathbf{A}_{N \times L}$ . Equation 6 calculates the number of multiply accumulate (MAC) operations required to calculate  $\mathbf{R}_{L \times L}$  for a single image cube.

$$MAC_{HSI} = NL^2 \quad (6)$$

Data preprocessing prior to HSI analysis adds to the system’s processing requirements. First, the raw data from the HSI sensor is preprocessed to correct for defects common in image sensors. Specifically, each value in the

image cube must be offset to account for readout noise and dark current and then scaled to adjust for flat field effects. Since the operations per value are roughly equivalent to a single MAC operation, and there are  $N \times L$  values for each image cube, raw data preprocessing requires  $L$  times less computation than HSI analysis. Since  $L > 100$  for most HSI systems, the raw data preprocessing resource demands are negligible.

We analyze two HSI sensors: the Hyperion [15], [16] on the Earth Observing-1 (EO-1) satellite, which orbits the Earth at about 6.7 km/s in a low earth orbit (LEO) at a 705 km altitude and the Airborne Visible / Infrared Imaging Spectrometer (AVIRIS) [17], which has been flown on four different aircraft platforms, but our analysis focuses on NASA’s ER-2 jet aircraft platform that travels at approximately 203 m/s at an altitude of 20 km. As shown in Figure 9, both sensors travel at high altitudes, capturing single lines of pixels at a time. These lines are perpendicular to the path of the sensor, and the combination of many



**Figure 9 – HSI sensors capture data one line at a time, forming an image cube from many adjacent lines as the sensor moves forward [17]**

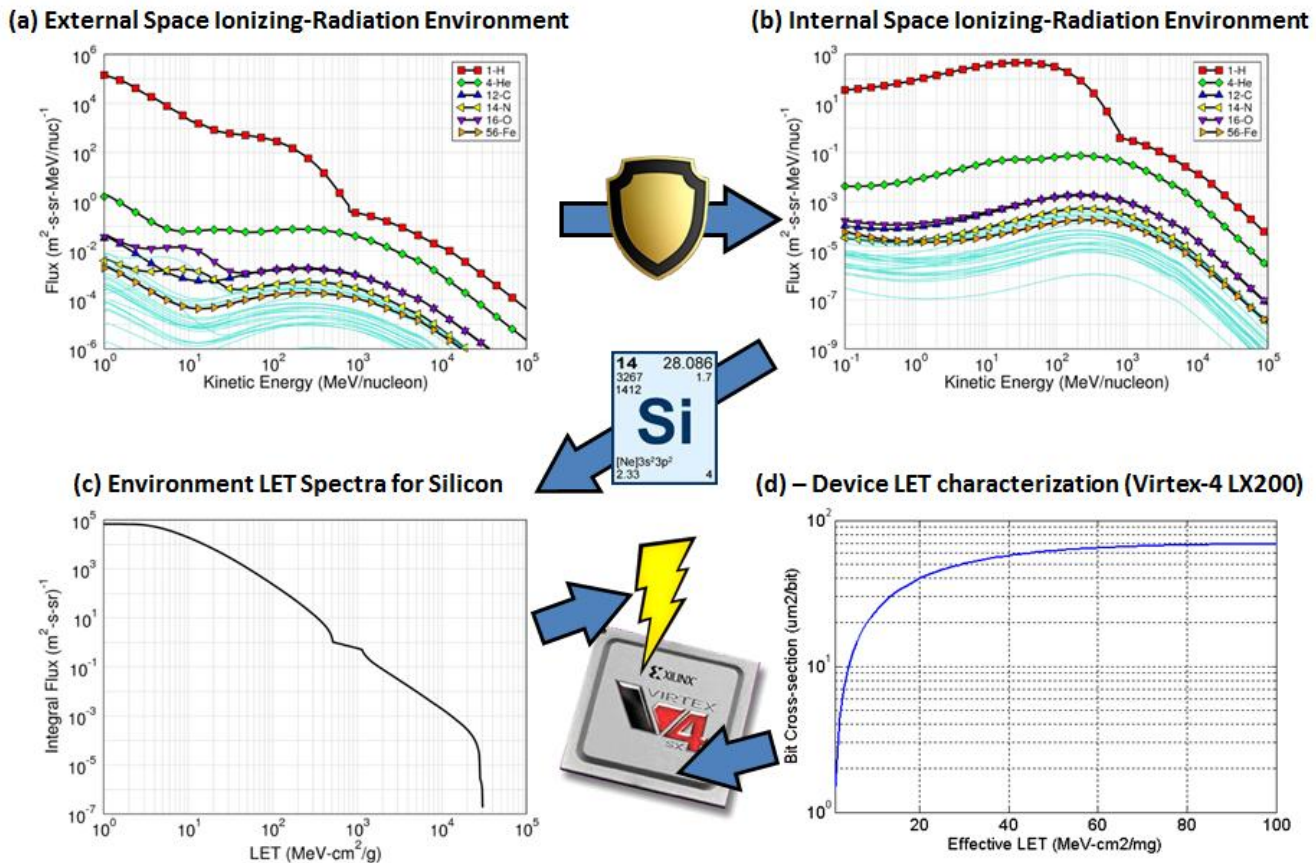
adjacent lines forms an image cube. The Hyperion captures an image every 2.95 seconds and produces an image cube 256 pixels wide, 660 lines long, and 196 12-bit spectral bands deep, requiring a total of 2.2 billion 32-bit integer MAC operations per second (OPS). The AVIRIS captures an image every 50 seconds and produces an image cube 677 pixels wide, 512 lines long, and 224 spectral bands deep, requiring a total of 348 million 32-bit integer MAC OPS.

#### 5-2. Framework Evaluation Metric Calculation

In order to clearly define our framework’s methodologies and contributions, this subsection details the power and dependability evaluation metric calculation process for the Hyperion and AVIRIS case-study missions using a Virtex-4 LX200 device with the ABFT FT strategy.

The Virtex-4 LX200 device is the largest device in the 90 nm Virtex-4 family and features 51.3 million configurable bits. Table 1 details the device’s characteristics. The device has a CD of 20.9 billion 32-bit integer MAC OPS. The





**Figure 10 – CREME96 uses environmental (a), shielding (b), and material (c) data to model environment and calculate upset rates based upon device radiation response data (d)**

device’s minimum (0% utilization) and maximum (100% utilization) power consumptions are 1.27 and 12.9 Watts, respectively.

**Table 1 – Virtex-4 LX200 properties**

Configurable Bits	32-bit Int MAC OPS	Min Power	Max Power
51,368,584	20.9 billion	1.27 W	12.905 W

*EO-1 Hyperion*—The required processing for the EO-1 Hyperion mission is 2.2 billion 32-bit integer MAC OPS (Section 5-1), resulting in a 10.5% device utilization for the Virtex-4 LX200. Pessimistically assuming 10% overhead [18] for the ABFT FT strategy produces a device-FT utilization of 11.6%. Finally, linear interpolation between the device’s power range (Equation 2) reveals a total power consumption of 2.62 Watts.

The EO-1 Hyperion mission operates in space, where the primary radiation concerns are trapped protons and heavy ions. Most trapped protons originate from the Sun’s solar winds and are trapped by the Earth’s magnetosphere, whereas heavy ions are highly charged particles originating from outside of the solar system. Both radiation hazards are reduced by increased solar activity, which causes

atmospheric expansion to remove low-orbiting trapped protons and stronger solar winds to repel heavy ions entering the solar system.

CREME96 calculates the effects of these particles on processing devices by reporting the expected upset rate for a device in a given orbit. We use the NORAD two-line element (TLE) [19] for EO-1 to supply the orbit parameters, and the solar-minimum model to ensure the dependability metric is accurate for the worst case. From these parameters, CREME96 creates a model of the external space ionizing-radiation environment similar to Figure 10a, which models the flux of various particle types and energies around the EO-1. After we specify a typical shielding of 100 mils of aluminum, CREME96 creates a similar model for the radiation environment inside the EO-1 as depicted in Figure 10b. From the internal radiation model, CREME96 models the LET spectra for silicon, which shows the particle flux per LET value as in Figure 10c. Finally, Figure 10d shows the Virtex-4 LX200 heavy-ion and trapped-proton LET curves from the device radiation response data that CREME96 uses to determine the upset rate of the device on the EO-1 platform. Table 2 shows the heavy-ion and trapped-proton Weibull parameters for CREME96. The heavy ion-induced upset rate is 263.6 upsets per day and the trapped proton-induced upset rate is 4.24 upsets per day, for

a total device upset rate of 267.8 upsets per day. A device utilization of 10.5% results in an effective device upset rate of 28.1 upsets per day.

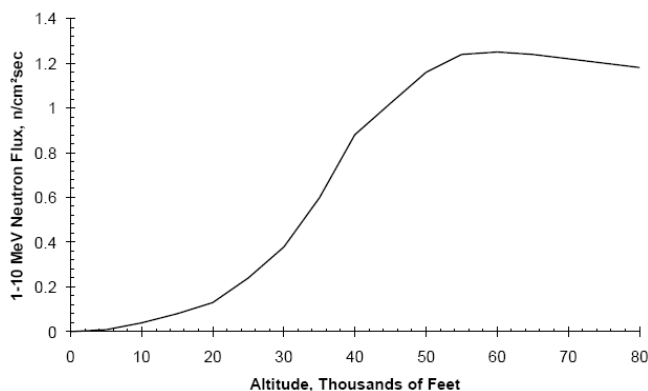
**Table 2 – Virtex-4 CREME96 Weibull parameters [20]**

Particle	Onset	Width	Power	Limit
Heavy Ion	0.5 $\frac{\text{MeV}\cdot\text{cm}^2}{\text{mg}}$	30 $\frac{\text{MeV}\cdot\text{cm}^2}{\text{mg}}$	1.3	70 $\frac{\mu\text{m}^2}{\text{bit}}$
Trapped Proton	4 MeV	80 MeV	0.586	0.0152 $\frac{10^{-12}\text{cm}^2}{\text{bit}}$

Since the ABFT FT strategy adds 10% overhead, the upset rate increases to 30.9 upsets per day. We also assume a pessimistic 90% coverage [18] for the ABFT detection. In the event of an upset detection, processing on the current image cube is restarted, resulting in no overall adverse effects for the system as long as there are no impending, hard real-time deadlines. With 90% coverage, the effective device upset rate drops to 3.09 upsets per day, which is equivalent to an MTBF of 7.77 hours.

*ER-2 AVIRIS*—The required processing for the ER-2 AVIRIS mission is 348 million 32-bit integer MAC OPS (Section 5-1), resulting in a 1.67% device utilization for the Virtex-4 LX200. Pessimistically assuming 10% overhead for the ABFT FT strategy produces a device-FT utilization of 1.83%. Finally, a linear interpolation between the device’s power range (Equation 2) reveals a total power consumption of 1.48 Watts.

The ER-2 AVIRIS mission operates in the Earth’s atmosphere at an altitude of 20 km, where the primary radiation concern is cascading neutrons. Energetic primary cosmic ray particles above 1 GeV constantly enter Earth’s atmosphere, collide with atmospheric particles, and release energy in the form of many secondary particles. These secondary particles then cascade into more energetic particles as they continue to collide with the atmosphere. By 20 km, almost all primary particles have converted into secondary particles, and atmosphere quickly absorbs most secondary particles. However, neutrons do not react as easily with other particles, so neutrons continue down through the atmosphere. As shown in Figure 11, there is a



**Figure 11 – Neutron flux vs. altitude, peaks at 60,000 ft [21]**

peak flux of 1.24  $\frac{\text{neutron}}{\text{cm}^2\cdot\text{s}}$  at 60,000 ft and only 0.0031  $\frac{\text{neutron}}{\text{cm}^2\cdot\text{s}}$  at ground level. Table 3 shows that lower solar activity and increased distance from the equator also result in higher neutron flux.

We assume a high flux during a solar minimum and polar latitude to ensure the dependability metric is accurate for the worst case. Since the ER-2 flies at 20 km, 60,000 ft is an appropriate altitude for estimating neutron flux. Table 3 reports that the worst-case neutron flux for the ER-2 is 24,859  $\frac{\text{neutron}}{\text{cm}^2\cdot\text{hr}}$ , and the device radiation response data for the Virtex-4 LX200 reports a cross-section of  $1.55\times 10^{-14} \frac{\text{cm}^2}{\text{bit}}$ . Multiplying the flux and cross-section produces an upset rate of  $3.85\times 10^{-10}$  upsets per bit-hour or 0.475 upsets per device-day. A device utilization of 1.67% results in an effective device upset rate of  $7.93\times 10^{-3}$  upsets per day. ABFT calculations similar to those used for the EO-1 Hyperion mission produce a final MTBF of  $1.15\times 10^{-3}$  days.

**Table 3 – Neutron flux vs. altitude and location [22]**

Latitude	Solar Activity	Flux Ranges ( $\frac{\text{neutron}}{\text{cm}^2\cdot\text{hr}}$ )		
		10,000 ft	30,000 ft	60,000 ft
Equator	Active	54-66	609-806	1776-2533
	Quiet	57-72	667-907	2038-3020
45°	Active	105-141	1604-2723	6362-15016
	Quiet	121-178	1969-3832	8541-24261
Polar	Active	142	2744	15286
	Quiet	179	3884	24859

**5-3. Results and Analysis**

For both missions, our framework computes the power and dependability evaluation metrics for 6 devices and 3 FT strategies, resulting in a total of 18 designs. We evaluate three of the most recent Virtex families (Virtex-4, Virtex-5, and Virtex-6), two low-power Spartan families (Spartan-3 and Spartan-6), and the radiation-hardened Xilinx SIRF device. For consistency, we choose the largest device from each family. The FT strategies include no fault tolerance (NFT), ABFT, and TMR. The TMR method assumes that voting takes place at the completion of each image cube.

Literature research provides space radiation response data for the Virtex-4 [23], Virtex-5 [23], Spartan-3 [24] [25], and SIRF [26]. Xilinx also provides neutron radiation response data for all of the devices except the SIRF [27]. Although the researched devices do not exactly match the devices we selected to study, we expect the radiation responses within a device family to be nearly identical since all of the devices within a family share the same bit-level structure and it is reasonable to reuse radiation response data for all devices within a family after adjusting for the devices’ number of configuration bits. Additionally, most of the trapped proton data only shows results for 63 MeV. For these cases, the Weibull parameters in Table 2 provide a reasonable estimate once the limit is set to 180% of the provided 63 MeV value.

EO-1 Hyperion Satellite Mission						
Device Fam.	Virtex-4	Virtex-5	Virtex-6	Spartan-3	Spartan-6	SIRF
Part Num	LX200	LX330T	LX760	5000	LX150T	5QV
Power Consumption (Watts)						
NFT	2.5	4.0	4.9	1.0	0.61	7.6
ABFT	2.6	4.0	5.0	1.1	0.66	7.7
TMR	4.9	5.1	5.9	2.4	1.6	9.2
Dependability MTBF (Days)						
NFT	3.5E-02	3.1E-01	<u>9.5E-02</u>	1.1E-01	<u>7.8E-02</u>	6.8E+02
ABFT	3.2E-01	2.8E+00	<u>8.6E-01</u>	1.0E+00	<u>7.1E-01</u>	6.2E+03
TMR	1.2E+01	8.8E+02	<u>8.5E+01</u>	1.2E+02	<u>5.7E+01</u>	5.7E+09

ER-2 AVIRIS Aircraft Mission						
Device Fam.	Virtex-4	Virtex-5	Virtex-6	Spartan-3	Spartan-6	SIRF
Part Num	LX200	LX330T	LX760	5000	LX150T	5QV
Power Consumption (Watts)						
NFT	1.5	3.5	4.5	0.45	0.18	7.0
ABFT	1.5	3.5	4.5	0.46	0.18	7.0
TMR	1.9	3.7	4.7	0.66	0.34	7.3
Dependability MTBF (Days)						
NFT	1.3E+02	5.3E+02	2.2E+02	1.9E+02	3.8E+02	3.9E+05
ABFT	1.1E+03	4.8E+03	2.0E+03	1.7E+03	3.4E+03	3.5E+06
TMR	9.2E+06	1.6E+08	2.8E+07	2.0E+07	8.2E+07	8.6E+13

Figure 12 – Power and dependability metrics for all combinations of six devices, three FT strategies, and two HSI missions

The heavy ion Weibull parameters in Table 2 are also used when only the heavy ion limit is available.

To the best of our knowledge, there is no space radiation data publicly available for the Virtex-6 and Spartan-6. To predict the heavy-ion and trapped-proton responses for the Virtex-6, we perform a linear regression on radiation data for the Virtex, Virtex-2, Virtex-4, and Virtex-5 to find a trend between Virtex family feature size and limiting cross-section. The Spartan-6 radiation data is found by adjusting this trend based on the relationship between the 90 nm Spartan-3 family and the 90 nm regression Virtex data.

Since we do not have access to SIRF tools, we estimate the SIRF power range and CD by analyzing the Virtex-5 FX130T, which is logically identical to the SIRF. Xilinx documents specify a block memory maximum frequency of 360 MHz for the SIRF [26] and 550 MHz for the Virtex-5 [28]. Since the Virtex-5 FX130T CD is bandwidth-limited, we assume the SIRF’s CD is equal to 65.45% of the FX130T CD. For increased fault-mitigating capabilities, the SIRF’s configuration memory cells double in transistor count. Due to the SIRF’s extra logic, we calculate an increase over the FX130T of 100% for static power and 30.9% for dynamic power after adjusting for the reduced clock rate. Finally, we estimate neutron radiation response

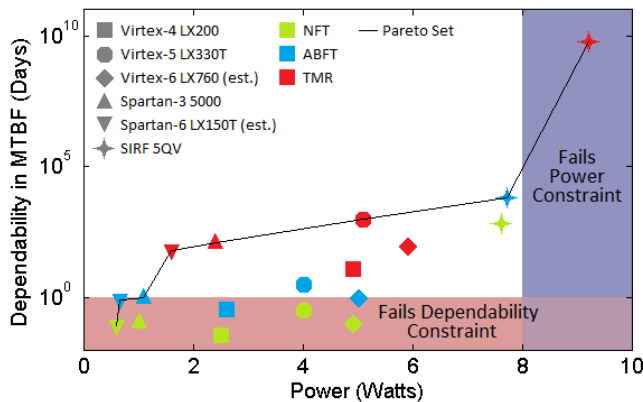


Figure 13 – Successful Pareto-optimal designs for EO-1 Hyperion mission

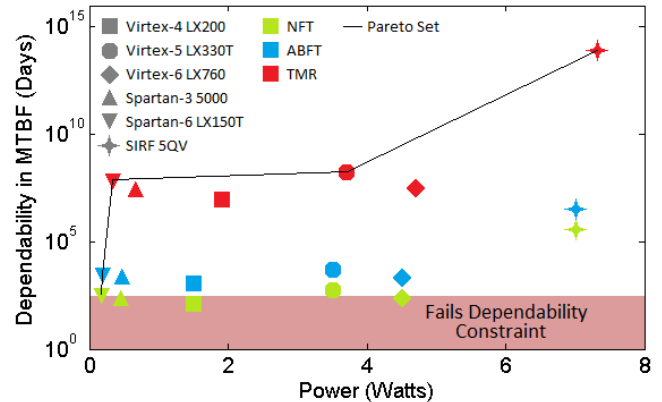


Figure 14 – Successful Pareto-optimal designs for ER-2 AVIRIS mission

data for the SIRF by assuming that the SIRF's fault-mitigating capabilities apply to neutron-induced upsets just as well as to heavy ion-induced upsets.

Figure 12 depicts the power and dependability metrics for all 18 designs for both the EO-1 Hyperion and ER-2 AVIRIS missions. Results based on estimates using feature size linear regression are shown in underlined italics.

Figures 13 and 14 demonstrate how our framework determines the Pareto-optimal set of designs for the EO-1 Hyperion and ER-2 AVIRIS missions, respectively. For illustrative purposes, the design constraints for the EO-1 Hyperion mission are a power consumption less than eight Watts and an MTBF greater than one day. Similarly, for example, the design constraints for the ER-2 AVIRIS mission are a power consumption less than 15 Watts and an MTBF greater than one year.

For the EO-1 Hyperion mission, the framework selects ABFT on Spartan-3, TMR on Spartan-6, TMR on Spartan-3, TMR on Virtex-5, and ABFT on SIRF as the final design set for designer evaluation. NFT on Spartan-6, ABFT on Spartan-6, and TMR on SIRF designs are Pareto-optimal and would be in the final design set, but these designs either consume too much power or are not sufficiently dependable. For each of the other successful unselected designs (non-Pareto optimal), at least one of the final designs is superior to the unselected design in both power and dependability. Therefore, the framework only presents designers with five final designs, since there is no advantage in selecting any of the other designs.

For the ER-2 AVIRIS mission, the framework selects NFT on Spartan-6, ABFT on Spartan-6, TMR on Spartan-6, TMR on Virtex-5, and TMR on SIRF as the final design set. The required processing for the ER-2 AVIRIS is roughly six times less than that of the EO-1 Hyperion, resulting in an average device utilization of 1.3%. Since static power is responsible for almost all of the consumed power, even tripling the dynamic power with high-overhead FT strategies like TMR has only a very modest effect on total power consumed. Therefore, TMR is a desirable FT strategy for the ER-2 AVIRIS mission, since designers can include TMR with only a small increase in power. The only reason to not include TMR would be for ultra-low power consumption, which would favor ABFT or NFT on Spartan-6 designs.

## 6. CONCLUSIONS

In this paper, we have introduced a novel framework that leverages past research and successes in device, application, and fault-tolerant (FT) strategy analysis to aid in the design of on-board processing systems. When supplied with a designer-defined mission and application, our framework analyzes a database composed of literature research and experimental data to provide designers with a final set of Pareto-optimal device/FT strategy designs. The

framework's evaluation metrics allow designers to select the best design from this final set depending on desired tradeoffs.

To illustrate the effectiveness of our framework, we analyzed the on-board processing potential of two currently deployed HSI missions. Our framework evaluated all combinations of six Xilinx FPGAs and three FT strategies for a total of 18 unique designs. For both missions, five final optimal designs were selected, which ranged from very-low power to very-high dependability. We verified our framework's success based on the framework's ability to reduce the design space search from 18 designs to a simple tradeoff decision between five designs.

Future work involves several expansions to our framework. A new Realizable Utilization (RU) methodology for device comparison purposes reports the amount of performance a typical designer is able to realize out of a device for a certain application as compared to the device's reported capability. RU enhances our framework by the evaluation of a device's CD with respect to an application. We also plan to include fault-injection analysis into our dependability metric calculation process, which would provide greater insight into the true vulnerability of certain applications and the behavior of various FT strategies. Research into device total ionizing dose may also lead to a new metric to evaluate the expected lifetime of a mission.

## ACKNOWLEDGMENTS

This work was supported in part by the I/UCRC Program of the National Science Foundation under Grant No. EEC-0642422. The authors gratefully acknowledge vendor equipment and tools provided by Xilinx that helped make this work possible.

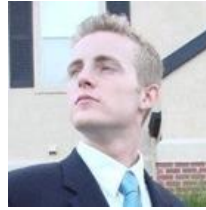
## REFERENCES

- [1] Zitzler, E.; Thiele, L.; , "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *Evolutionary Computation, IEEE Transactions on* , vol.3, no.4, pp.257-271, Nov 1999
- [2] Pease, R.L.; Johnston, A.H.; Azarewicz, J.L.; , "Radiation testing of semiconductor devices for space electronics ," *Proceedings of the IEEE* , vol.76, no.11, pp.1510-1526, Nov 1988.
- [3] J. Williams, A. George, J. Richardson, K. Gosrani, and S. Suresh, "Computational Density of Fixed and Reconfigurable Multi-Core Devices for Application Acceleration," *Proc. of Reconfigurable Systems Summer Institute*, Urbana, IL, July 7-10, 2008.
- [4] Tylka, A.J.; Adams, J.H., Jr.; Boberg, P.R.; Brownstein, B.; Dietrich, W.F.; Flueckiger, E.O.; Petersen, E.L.; Shea, M.A.; Smart, D.F.; Smith, E.C.; , " CREME96: A

- Revision of the Cosmic Ray Effects on Micro-Electronics Code," *Nuclear Science, IEEE Transactions on* , vol.44, no.6, pp.2150-2160, Dec 1997.
- [5] Francis P. Mathur , Algirdas Avižienis, Reliability analysis and architecture of a hybrid-redundant digital system: generalized triple modular redundancy with self-repair, Proceedings of the May 5-7, 1970, spring joint computer conference, May 05-07, 1970, Atlantic City, New Jersey.
- [6] Xilinx. (2003, August 1). *XAPP383 Single Error Correction and Double Error Detection (SECCDED) with CoolRunner-II<sup>TM</sup> CPLDs (v1.1)* [Online]. Available: <http://www.xilinx.com>
- [7] Kuang-Hua Huang; Abraham, J.A.; , "Algorithm-Based Fault Tolerance for Matrix Operations," *Computers, IEEE Transactions on* , vol.C-33, no.6, pp.518-528, June 1984.
- [8] A. Jacobs, A. George, and G. Cieslewski, "Reconfigurable Fault Tolerance: A Framework for Environmentally Adaptive Fault Mitigation in Space," *Proc. of Intl. Conference on Field-Programmable Logic and Applications*, Prague, Czech Republic, Aug. 31 - Sep. 2, 2009.
- [9] S. Yousuf and A. Gordon-Ross, "Partially Reconfigurable System-on-Chips for Adaptive Fault Tolerance," *Proc. of IEEE International Conference on Field-Programmable Technology (FPT)*, Dec. 12-14, 2011.
- [10] K. Asanovic et al., The Landscape of Parallel Computing Research: A View from Berkeley, tech. report UCB/EECS-2006-183, Dept. Electrical Eng. and Computer Science, Univ. of Calif., Berkeley, 2006.
- [11] Aerospace. *Single Event Effects Testing Primer* [Online]. Available: <http://www.aero.org>
- [12] G. Vane, *High spectral resolution remote sensing of the Earth*, Sensors, no. 2, pp. 11–20, 1985.
- [13] Clark, R.N., King, T.V.V., Ager, C., and Swayze, G.A., 1995, Initial vegetation species and senescence/stress mapping in the San Luis Calley, Colorado using imaging spectrometer data. *Proceedings: Summitville Forum '95*, H.H. Posey, J.A. Pendelton, and D. Van Zyl Eds., Colorado Geological Survey Special Publication 38, p. 64-69.
- [14] NASA Jet Propulsion Laboratory. *HyspIRI Mission Study* [Online]. Available: <http://hyspiri.jpl.nasa.gov>
- [15] Pearlman, J.; Carman, S.; Segal, C.; Jarecke, P.; Clancy, P.; Browne, W.; , "Overview of the Hyperion Imaging Spectrometer for the NASA EO-1 mission," *Geoscience and Remote Sensing Symposium, 2001. IGARSS '01. IEEE 2001 International* , vol.7, no., pp.3036-3038 vol.7, 2001.
- [16] Pearlman, J.S.; Barry, P.S.; Segal, C.C.; Shepanski, J.; Beiso, D.; Carman, S.L.; , "Hyperion, a space-based imaging spectrometer," *Geoscience and Remote Sensing, IEEE Transactions on* , vol.41, no.6, pp. 1160- 1173, June 2003.
- [17] NASA Jet Propulsion Laboratory. *AVIRIS Airborne Visible / Infrared Imaging Spectrometer* [Online]. Available: <http://aviris.jpl.nasa.gov/aviris>
- [18] Silva, J.G.; Prata, P.; Rela, M.; Madeira, H.; , "Practical issues in the use of ABFT and a new failure model," *Fault-Tolerant Computing, 1998. Digest of Papers. Twenty-Eighth Annual International Symposium on* , vol., no., pp.26-35, 23-25 Jun 1998
- [19] Celestrak. (2011, November 2). *NORAD Two-Line Element Sets Current Data* [Online]. Available: <http://celestrak.com/NORAD/elements/resource.txt>
- [20] G. R. Allen, G. Swift, and C. Carmichael, "Virtex-4QV Static SEU Characterization Summary," <http://parts.jpl.nasa.gov/docs/NEPP07/NEPP07FPGA4Static.pdf>
- [21] Taber, A.; Normand, E.; , "Single event upset in avionics," *Nuclear Science, IEEE Transactions on* , vol.40, no.2, pp.120-126, Apr 1993
- [22] Quinn, H.; Manuzzato, A.; Fairbanks, T.; Dallmann, N.; DesGeorges, R.; "High-Performance Computing for Airborne Applications," *Nuclear and Space Radiation Effects Conference 2010*, Denver, CO, July 19-23, 2010.
- [23] Quinn, H.M.; Graham, P.S.; Wirthlin, M.J.; Pratt, B.; Morgan, K.S.; Caffrey, M.P.; Krone, J.B.; , "A Test Methodology for Determining Space Readiness of Xilinx SRAM-Based FPGA Devices and Designs," *Instrumentation and Measurement, IEEE Transactions on* , vol.58, no.10, pp.3380-3395, Oct. 2009.
- [24] Manuzzato, A.; Gerardin, S.; Paccagnella, A.; Sterpone, L.; Violante, M.; , "On the Static Cross Section of SRAM-Based FPGAs," *Radiation Effects Data Workshop, 2008 IEEE* , vol., no., pp.94-97, 14-18 July 2008.
- [25] Hiemstra, D.M.; Chayab, F.; Szajek, L.; , "Dynamic single event upset characterization of the Virtex-II and Spartan-3 SRAM field programmable gate arrays using proton irradiation," *Radiation Effects Data Workshop, 2004 IEEE* , vol., no., pp.79-84, 22-22 July 2004.
- [26] Xilinx. (2011, July 11). *DS192 Radiation-Hardened, Space-Grade Virtex-5QV Device Overview (v1.2)* [Online]. Available: <http://www.xilinx.com>
- [27] Xilinx. (2011, August 2). *UG116 Device Reliability Report Second Quarter 2011 (v7.0)* [Online]. Available: <http://www.xilinx.com>

[28] Xilinx. (2009, February 6). *DS100 Virtex-5 Family Overview* (v5.0) [Online]. Available: <http://www.xilinx.com>

## BIOGRAPHIES



**Nicholas Wulf** is a doctoral student in ECE at the University of Florida. He is group leader and research assistant in the advanced processing devices group in the NSF CHREC Center at Florida. His research interests include analysis and comparison of fixed and reconfigurable device architectures and low-overhead fault-tolerant techniques.



**Alan D. George** is Professor of ECE at the University of Florida, where he serves as Director of the NSF Center for High-performance Reconfigurable Computing known as CHREC. He received the B.S. degree in CS and M.S. in ECE from the University of Central Florida, and the Ph.D. in CS from the Florida State University. Dr. George's research interests focus upon high-performance architectures, networks, systems, services, and applications for reconfigurable, parallel, distributed, and fault-tolerant computing. He is a senior member of IEEE and SCS, and a member of ACM and AIAA.



**Ann Gordon-Ross (M'00)** received her B.S and Ph.D. degrees in Computer Science and Engineering from the University of California, Riverside (USA) in 2000 and 2007, respectively. She is currently an Assistant Professor of ECE at the University of Florida and is a member of the NSF Center for High Performance Reconfigurable Computing (CHREC). She is also faculty advisor for the Women in Electrical and Computer Engineering (WECE) and the Phi Sigma Rho National Society for Women in Engineering and Engineering Technology. She received her CAREER award from the National Science Foundation in 2010 and Best Paper awards at the Great Lakes Symposium on VLSI (GLSVLSI) in 2010 and the IARIA International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM) in 2010. Her research interests include embedded systems, computer architecture, low-power design, reconfigurable computing, dynamic optimizations, hardware design, real-time systems, and multi-core platforms.